

Distributed Optimization for Shared State Systems: Applications to Decentralized Freeway Control via Subnetwork Splitting

Jack Reilly¹ and Alexandre M. Bayen²

Abstract—Optimal control problems on dynamical systems are concerned with finding a control policy which minimizes a desired objective, where the objective value depends on the future evolution of the system (the state of the system), which in turn depends on the control policy. For systems which contain subsystems that are disjoint across the state variables, distributed optimization techniques exist which iteratively update subsystems concurrently and then exchange information between subsystems. This article presents a method, based on the asynchronous ADMM algorithm, which extends these techniques to subsystems with shared control and state variables, while maintaining a similar communication structure. The method is used as the basis for splitting network flow control problems into many subnetwork control problems with shared boundary conditions. The decentralized and parallel nature of the method permits high scalability with respect to the size of the network. The method is applied to decentralized, coordinated ramp metering and variable speed limit control on a realistic freeway network model using distributed model predictive control.

Index Terms—distributed algorithms, traffic control

I. INTRODUCTION

Finite-horizon optimal control is a popular method for computing predictive control strategies for dynamical systems [1], [2], its suitability and effectiveness growing with the increase of computational power and pervasiveness of physical sensing. In general, a finite-horizon optimal control problem will take the following form:

$$\min_{x \in X} f(s, x) \quad (1)$$

$$\text{subject to: } s = g(x) \quad (2)$$

where x represents the vector of control variables belonging to the set of feasible controls X (which we may assume to be \mathbb{R}^n for simplicity), s represents the vector of “state” variables, constrained to be a deterministic function $g(x)$ of the control, and f is some objective function of the control and state we wish to minimize.

Distributed Optimization and Control: Much attention has recently been given to distributed methods for finite-horizon optimal control problems, where g is assumed to be linear and f is assumed to be quadratic or convex.

*This work was supported by the California Department of Transportation under the Connected Corridors program

¹Jack Reilly is a Ph.D student of Civil and Environmental Engineering University of California, Berkeley, 652 Sutardja Dai Hall, Berkeley CA 94720, US jackdreilly@berkeley.edu

²Alexandre M. Bayen is a Chancellor Professor of Electrical Engineering and Computer Sciences and Civil and Environmental Engineering University of California, Berkeley, 652 Sutardja Dai Hall, Berkeley CA 94720, US bayen@berkeley.edu

Distributed optimization has been found useful for at least two reasons. Firstly, the parallelizability of the individual sub-problems allows for faster computation time and better overall convergence properties [3], [4], [5], [6], [7]. Secondly, physical systems often have actuation points physically distributed in space, creating a need for distributed control algorithms which limit the required shared information and communication between subsystems [8], [9], [10].

Different assumptions on the structure, smoothness, and convexity of f , X and g leads to different convergence bounds and communication bounds. In optimal control, a method presented in [3] for decoupling the quadratic terms from the nonquadratic terms leads to efficient caching techniques shown to be effective in FPGA applications. A distributed gradient descent-based approach is given in [9], which has $O\left(\frac{1}{\sqrt{k}}\right)$ convergence to the global optimum in the general case, where k is the number of iterations of the algorithm. A common dual-decomposition technique employed for distributed optimal control is the *alternating directions method of multipliers* [11], [12], [3] (ADMM), which has been shown to have $O\left(\frac{1}{k}\right)$ convergence under certain assumptions of the decomposability of the objectives [13]. Additionally, an accelerated version of ADMM, based on Nesterov’s algorithm [14] can give $O\left(\frac{1}{k^2}\right)$ convergence when the decomposed objectives are smooth [5].

When the coupling between systems takes on some sparse form, then one can devise algorithms with limited communication, which can be beneficial from an architectural standpoint and a convergence standpoint [15]. Optimal control problems where subsystems have disjoint state variables and coupled control variables have been shown to be amenable to decomposition techniques for distributed optimization [6], [9]. [8] shows how ADMM decomposition leads to less communication without a decrease in solution accuracy.

In [8], [6], [9], the subsystems with disjoint state are modeled as agents tasked with optimizing over their own subsystem, where agents which share some control variables are connected by some edge in a communication graph. Thus, the more sparse the coupling of systems, the lesser the communication requirements. Such a model is referred to as *multi-agent optimization* [13]. In systems with coupling due to physical proximity, this consequence has the added benefit of requiring only physically local communication, and removes the need for any centralized controller or hub for communication. In [13], an asynchronous form of ADMM (subsequently referred to as A-ADMM) is presented for multi-agent optimization, which permits agents to update

themselves in arbitrary order, with communication only required between neighboring agents. The method in [13] does not present an accelerated version and is shown to have $O(\frac{1}{k})$ convergence.

Subsystems with Coupled State: One recurring assumption in the distributed optimization literature above is that subsystems have disjoint state variables. For network flow problems, where subsystems correspond to partitions of a network into subnetworks, such an assumption does not hold. To see this, one can imagine a traffic light timing plan causing a traffic jam which spreads across the entire freeway network [16] or a bottleneck of planes in an airspace affecting flight times throughout the air network [1]. In these examples, the state at the boundaries of subnetworks clearly influence the state of neighboring subnetworks. As a result, it is not possible to decompose the subsystems by only sharing control parameters without coupling each subsystem to all control variables and modeling the evolution of the entire network within each subsystem.

Yet, freeway traffic and air traffic subsystems have a very sparse coupling in their state variables. For instance, discrete traffic models [17], [18] often assume that the speed of traffic on a particular section of road is only a function of the speed of traffic on neighboring links. Thus, each subnetwork subsystem would only share a small number of control variables and state variables with other subsystems, precisely those which physically share a border with the subsystem.

The first main contribution of this article is a multi-agent optimization algorithm which exploits the sparsity of such systems. Based on A-ADMM [13], the algorithm permits each agent (subsystem) to share both control and state variables with neighboring agents, while still converging to produce the globally optimal control, given the standard assumption of convex objectives and linear constraints. At a high level, the algorithm “relaxes” the state variables *external* to an agent while constraining *internal* state variables to adhere to the subsystem’s dynamics. Since A-ADMM eventually brings all shared variables between agents into *consensus* (i.e. the difference between shared variables converges to zero), the relaxed external state variables will converge to satisfying the original constraints.

The other main contribution of this article is an asynchronous and decentralized ramp-metering and variable-speed-limit (VSL) control scheme which leverages the shared-state A-ADMM algorithm. The proposed control scheme fits naturally in the context of distributed transportation management agencies, where no central control is required, subnetworks may communicate in an ad-hoc, asynchronous fashion, while still achieving globally optimal objectives (i.e. collaboration is a social optimum, not a Nash equilibrium).

The rest of the article is structured as follows. Section II presents the general problem of posing a multi-agent optimal control problem, with the additional assumption that an agent may share both state and control variables with other agents. The problem is then posed in a form amenable to using the

A-ADMM algorithm in Section III. A systematic approach to modeling an optimal control problem over a dynamical network as a multi-agent distributed optimization problem is given in Section IV, as well as a discussion on the suitability of the method for scaling model predictive control on dynamical networks and potentially improving convergence [15]. We then present the application of distributed, predictive ramp-metering and VSL control on freeway networks in Section V followed by numerical results in Section VI with comparisons to existing distributed approaches. We conclude with some final remarks in Section VII.

Notation: For a vector x , let $x[i]$ be the i ’th element of x , and similarly let $y[i, j]$ be the element of the two-dimensional array in the i -th row and j -th column. Let $\text{card}(x)$ be the cardinality of a vector x , i.e. the number of elements in x . If we have a vector x with $\text{card}(x) = N$ and w a subset of $\{1, \dots, N\}$, then let x_w denote the vector selecting only those elements $x[i]$ where $i \in w$. We define the *concatenation* of vectors $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$ as the resulting vector $z \in \mathbb{R}^{n+m}$ constructed by appending the elements of y onto x . If a vector d is the concatenation $d = (a, b, c)$, then let $[d]_a$ be the sub-vector of d corresponding to the original element a .

II. PROBLEM STATEMENT

We wish to solve an optimization problem with a “free” global variable $x \in \mathbb{R}^n$ and a “dependent” variable $s \in \mathbb{R}^m$ which is a deterministic function of x , where n is the number of “control” parameters and m is the number of “state” parameters. We assume there is a partition of s into D disjoint subsets,

$$s = (s_{u(1)}, \dots, s_{u(D)}),$$

where $u(i)$ are subsets of $\{1, \dots, m\}$. Section IV discusses several strategies for choosing state partitions based on system dynamics. The objective function is assumed to be the sum of D sub-objectives, where sub-objective $f_i, i \in \{1, \dots, D\}$ is a convex function of only partition $s_{u(i)}$.¹ Furthermore, $s_{u(i)}$ is assumed to be a function of some subset of x and s . Explicitly, for each $i \in 1, \dots, D$, there is well-defined, linear function g_i and subsets $v(i)$ and $w(i)$ ($w(i) \cap u(i) = \emptyset$) where

$$s_{u(i)} = g_i((x_{v(i)}, s_{w(i)})). \quad (3)$$

The tuple $(x_{v(i)}, s_{w(i)})$ is the concatenation vector of $x_{v(i)}$ and $s_{w(i)}$. We omit the double parenthesis in the rest, for simplicity. One can view $u(i), v(i), w(i)$, as the *internal* state, the control, and the *external* state, respectively, of group i . We can now express the optimization problem we wish to solve as:

¹We omit the dependency of the objective on the control variable in this presentation for simplicity. It is still easy in this form to add control variables into the objective by duplicating a control variable into the state.

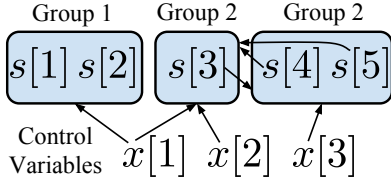


Fig. 1. Optimization problem partitioned into $D = 3$ disjoint state variable groups. Arrows allow us to compute the $u(i), v(i), w(i)$ subsets for each group i , where \rightarrow indicates functional dependency through Equation 3

TABLE I
SUBSETS $u(i), v(i), w(i)$ FOR FIGURE II EXAMPLE.

Group i	$u(i)$	$v(i)$	$w(i)$
$i = 1$	{1,2}	{1}	{}
$i = 2$	{3}	{1,2}	{4,5}
$i = 3$	{4,5}	{3}	{3}

$$\min_{x,s} \sum_{i=1}^D f_i(s_{u(i)}) \quad (4)$$

$$\text{subject to: } s_{u(i)} = g_i(x_{v(i)}, s_{w(i)}) \quad \forall i \in 1, \dots, D. \quad (5)$$

Figure II shows an example of how different sub-objectives may be coupled and Table I summarizes how one constructs the $u(i), v(i), w(i)$ subsets from the state and control coupling.

Dependency Graph: There are no assumptions on the subsets $v(i)$ and $w(i)$, which implies that the value of each sub-objective f_i is coupled to not just the sub-vector $s_{u(i)}$, but also the global variable x , and other sub-vectors $s_{u(j)}$. We can express this coupling as a dependency graph (V, E) , where vertices V are the sub-problems $i \in \{1, \dots, D\}$ and an edge $(i, j) \in E$ exists whenever

- 1) $w(i) \cap u(j) \neq \emptyset$ (g_j is a function of some variable in $s_{u(j)}$), **or**
- 2) $v(i) \cap v(j) \neq \emptyset$ (there is some $x[k]$ which both g_i and g_j depend upon).

Let the neighboring edges of node $i \in V$ be denoted by $E(i)$. A dependency graph construction for the example in Figure II is summarized in Table II. The intersection of subsets from Table I across different subproblems reveals that edges exist for groups (1, 2) and (2, 3), but not for (1, 3).

In Section III, we devise a distributed algorithm to solve Problem (4)-(5) with the following requirements:

- 1) Each processing node corresponds to a sub-objective node in the dependency graph.
- 2) Each node can be updated in parallel.
- 3) Each node i only exchanges information with its neighbors $E(i)$ in the dependency graph (V, E) .
- 4) The algorithm is asynchronous and decentralized, i.e. no central process is required and nodes can be updated arbitrarily.

III. ASYNCHRONOUS-ADMM ALGORITHM

We reformulate Problem (4)-(5) to permit a distributed solution method via A-ADMM. For each node $i \in V$, we

TABLE II
SUBSET INTERSECTION TERMS FOR FIGURE II EXAMPLE.

Edge (i, j)	$v(i) \cap v(j)$	$u(i) \cap u(j)$	$w(i) \cap u(j)$
$i, j = 1, 2$	{1}	{}	{}
$i, j = 2, 3$	{}	{3}	{4,5}
$i, j = 1, 3$	{}	{}	{}

duplicate the “shared variables” $x_{v(i)}$ and $s_{w(i)}$ as \bar{x}_i and \bar{s}_i respectively, and reformulate Problem (4)-(5) as:

$$\min_x \sum_{i=1}^D f_i(s_{u(i)}) \quad (6)$$

$$\text{subject to: } s_{u(i)} = g_i(\bar{x}_i, \bar{s}_i) \quad \forall i \in 1, \dots, D \quad (7)$$

$$\bar{s}_i = s_{w(i)} \quad \forall i \in 1, \dots, D \quad (8)$$

$$\bar{x}_i = x_{v(i)} \quad \forall i \in 1, \dots, D \quad (9)$$

The variable replication allows Constraint (7) in Problem (6)-(9) to be decoupled across nodes. To decouple Constraints (8) and (9), we follow a modified process from [13].

First, we duplicate each subset $s_{u(i)}$ with a vector s_i local to node $i \in V$, and then concatenate all local variables into a single variable $y_i = (s_i, \bar{x}_i, \bar{s}_i)$, such that y_i is restricted to the space:

$$Y_i = \{(s_i, \bar{x}_i, \bar{s}_i) : s_i = g_i(\bar{x}_i, \bar{s}_i)\}.$$

Finally, we can repose Constraints 2 and 3 in an *edge-wise* fashion as follows. For each edge $e = (i, j) \in E$, let $y_{i,e}$ and $y_{j,e}$ be the sub-vectors of y_i and y_j that are coupled through g_j and g_i , respectively. Then Problem (4)-(5) becomes:

$$\min_{(y_i \in Y_i)_{i \in V}} \sum_{i=1}^D f_i([y_i]_s) \quad (10)$$

$$\text{subject to: } y_{i,e} = y_{j,e} \quad \forall e \in E \quad (11)$$

By moving the edge constraints into the objective through a standard Lagrange multiplier approach, and adding a regularization term which is equal to zero for feasible solutions [12], we can construct the augmented Lagrangian \mathcal{L} formulation (with tunable augmenting coefficient ψ), and express the optimization problem as:

$$\min_{y=(y_i)_{i \in V}} \max_{\lambda=(\lambda_e)_{e \in E}} \mathcal{L}(y, \lambda) := \quad (12)$$

$$\sum_{i=1}^D f_i([y_i]_s) + \sum_{e \in E} \lambda_e^T (y_{i,e} - y_{j,e}) + \psi \|y_{i,e} - y_{j,e}\|_2^2, \quad (13)$$

The above form permits us to apply the A-ADMM algorithm as proposed and analyzed in [13], and shown in Algorithm 1. At a high-level, the algorithm iterates by first randomly selecting an edge $e = (i, j)$ from E . Then, nodes i and j update y_i and y_j respectively by minimizing the Lagrangian in Equation (12) in parallel, while holding all other variables $\{\lambda'_e\}_{e' \neq e}, \{y_k\}_{k \notin \{i, j\}}$ constant. The new y_i and y_j values

Algorithm 1 Asynchronous Edge Based ADMM

1: **while** Not Converged **do**
 2: Select edge $e = (i, j) \in E$
 3: **for** $q \in (i, j)$ **do**
 4:

$$y_q^{k+1} \leftarrow \arg \min_{y \in Y_q} f_q([y]_s) - \sum_{e \in E(q)} \Lambda_{q,e} \lambda_e^{k,T} (y_{q,e} - \bar{y}_e^k) + \frac{\psi}{2} \|y_{q,e} - \bar{y}_e^k\|_2^2$$

5: **end for**
 6: $\lambda_e^{k+1} \leftarrow \lambda_e^{k+1} - \frac{\psi}{2} (y_{i,e}^{k+1} - y_{j,e}^{k+1})$
 7: **for** $q \notin (i, j), e' \neq e$ **do**
 8: $y_q^{k+1} = y_q^k, \lambda_{e'}^{k+1} \leftarrow \lambda_{e'}^k$
 9: **end for**
 10: **end while**
 11: Note: $\tilde{y}_e^k = \frac{1}{2} (y_{i,e}^k + y_{j,e}^k)$
 12: Note: $\Lambda_{q,e} = \begin{cases} 1 & q = i \\ -1 & q = j \end{cases} \quad e = (i, j)$

are used to update the dual λ_e variables by applying a dual-ascent method [19]. Finally, the process is repeated *ad-infinitum* by updating a new edge selected from E , until some convergence or termination criteria are reached.

Remark. The equation in Line 4 differs slightly from the augmented Lagrangian in Equation (12) and is the result of a number of algebraic manipulations, which are explicitly derived in [12], [13].

Remark. We introduce the asymmetric coefficient $\Lambda_{q,e}$ to account for the fact that the terms for edge $e \in E(q)$ in Line 4 depend upon whether the updating problem q was the first or second term (i or j) in the edge pair.

IV. DISTRIBUTED OPTIMIZATION ON COUPLED DYNAMICAL SYSTEMS

Physical transport systems, such as freeway traffic networks [17], [20] or gas pipelines [21] are often naturally expressed as a network of individual dynamical systems which influence one another at contact points, or *junction points*. Given the coupling in dynamics across the entire network, optimizing over partitioned sub-systems, with no communication between systems, will lead to *greedy* solutions over the individual systems and sub-optimal global results [4]. Thus, any distributed, globally optimal control scheme applied to such systems must account for the *shared state* between the systems. We now show how this can be done using the multi-agent A-ADMM approach. Furthermore, we show how the algorithm naturally leads to a communication scheme which mirrors the physical structure of the underlying physical network.

Assume some discrete-time, discrete-space dynamical system which possesses a network-like dynamical coupling in space. Specifically, consider a graph (V^d, E^d) (not to be confused with the dependency graph (V, E) in Section II, where the d superscript is added to denote the *dynamical*

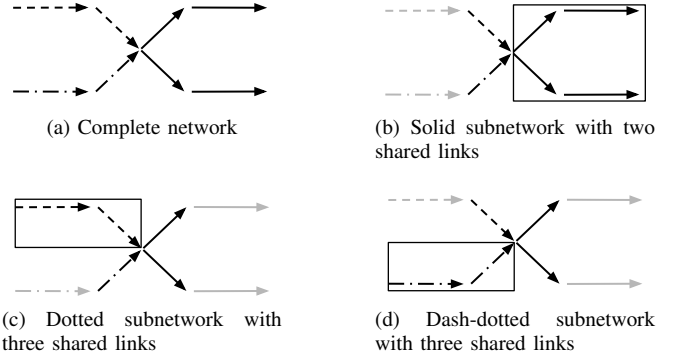


Fig. 2. A network is partitioned into three subnetworks: solid, dashed, and dash-dotted. Each subnetwork will share state with neighboring subnetworks. For a subnetwork i , the cells neighboring i , denoted by E_i^d , are shown in black, while those excluded from E_i^d are shown in gray.

network) where E^d represent the discrete-space *cells* and V^d are the *junction points* where cells connect to one another, i.e. each cell in E^d has a corresponding upstream and downstream junction both in V^d . Each discrete space “cell” $c \in \{1, \dots, N_d\}$ has for each discrete time step $k \in \{1, \dots, T_d\}$ both a control variable $x[c, k] \in \mathbb{R}$ and a state variable $s[c, k] \in \mathbb{R}$. The variable $s[c, k]$ is assumed to be a function of all state and control variables that satisfy two conditions:

- the time-step is $k - 1$, and
- the cell must share a junction with cell c .

Next, we wish to express a distributed optimization problem subject to the above dynamics in the form of Problem (4)-(5). To do so, we assume a partition of (V^d, E^d) into D subnetworks, which implies a partition of E^d into D subsets (E_1^d, \dots, E_D^d) and assume an objective f which is splittable across the state variables internal to each sub-network. This leads to a state partitioning $s = (s_{u(1)}, \dots, s_{u(D)})$, where $(c, k) \in u(i)$ iff $c \in E_i^d$. In practice, such partitionings may be chosen, for instance, to utilize a certain amount of computational power at the disposal of the designer, or may already have partitionings in place, such as regional transportation management districts.

Based on the two conditions for state dependencies above, we can deduce that the state of a sub-network depends on the control and state both internal to the sub-network and directly *neighboring* the sub-network. Explicitly, for sub-network i , we can express the dependent control variables as $x_{v(i)}$ where $(c, k) \in v(i)$ iff $c \in E_i^d$ or c neighbors a cell in E_i^d . Similarly, the shared state for sub-network i is $s_{w(i)}$, where $(c, k) \in w(i)$ iff $c \notin E_i^d$ and c neighbors a cell in E_i^d . Finally, we conclude that there exists some update equation g_i , specific to the particular dynamical system, where the constraint on $s_{u(i)}$ can be expressed familiarly as $s_{u(i)} = g_i(x_{v(i)}, s_{w(i)})$.

As an example, we can consider the network in Figure 2a, which is partitioned into three subnetworks based on line-style.

We see that four of the edges share a single junction

between the three subnetworks. Thus, the dynamics assumed above implies that each subnetwork will share state with each other subnetwork. Specifically, the solid-lined network in Figure 2b shares one cell each from the other two subnetworks, while the dashed and dash-dotted subnetworks in Figures 2c and 2d share two cells with the solid subnetwork and one cell with the opposite subnetwork. We note again that while each optimizing agent may have different values of the state on a particular cell in the network during intermediate stages of the A-ADMM algorithm, each copy of the state will eventually come into consensus as the shared-state A-ADMM algorithm converges.

Local Communication Requirements: At this point, all relevant parameters to Problem (4)-(5) have been specified. The assumption on the dynamical network coupling leads to a desirable dependency graph (V, E) for the system above. Since each sub-network only requires shared state from neighboring sub-networks in the sense of the *physical* network (V^d, E^d) , then the dependency graph (V, E) is constructed by assigning a sub-network to each node V and adding an edge (i, j) to E only for those sub-networks i and j which physically neighbor each other. Thus, the A-ADMM algorithm guarantees that communication only take place between physically neighboring systems. This is useful for situations where there are limitations in the networking capabilities due to physical distance, such as freeway traffic control systems, where collaborations may only exist for those districts near each other.

Furthermore, the formulation allows for a completely decentralized and asynchronous implementation of the global optimization problem. If, for instance, all nodes are managed by independent agencies with varying computational limits, then there are several practical benefits to the approach. For a single sub-network, since only information that is directly adjacent to other sub-networks needs to be shared with other sub-networks, much of the internal formulation of the sub-network can be made completely hidden from the larger network. The asynchronicity of the algorithm also permits for neighboring agencies to exchange information in an ad-hoc manner, and not be bottlenecked by slower updates between separate sub-networks. We also note that a synchronous version of our algorithm is easily achievable when preferable by requiring all edges to update once before a subsequent round of updates begin.

Scalability and Convergence of Subnetwork Splitting for Model Predictive Control: A common application of finite-horizon optimal control is in the context of model predictive control (MPC) [4], [16], where optimal control policies are recomputed in a *rolling-horizon* fashion. Given the optimal control problem beginning at a time-step t ,

$$\begin{aligned} \min_{x=\{x_t, \dots, x_{t+T}\}} \quad & f_t^{t+T}(s, x) \\ \text{subject to:} \quad & s = g_t^{t+T}(x), \end{aligned} \quad (14)$$

MPC chooses the control policy x_t to apply at time-step t by solving for $x = \{x_t, \dots, x_{t+T}\}$ in Equation (14) using a

prediction horizon of T and updating the objective f_t^{t+T} and constraints g_t^{t+T} based on the latest estimates of the initial conditions and boundary conditions.

In many MPC applications, the choice of a time-horizon is a trade-off between limiting computation time and accounting for far-future events. Yet, in an application such as freeway onramp metering, another factor in choosing an optimization time-horizon is the accuracy of the predictions of the boundary conditions, or specifically, anticipating future vehicle demands on freeway onramps. At some point, increasing the time-horizon will only decrease the effectiveness of the control due to the deviation in predicted model state versus reality. Thus, it is sometimes practical to consider the time-horizon fixed in MPC applications, at which point the scalability with respect to network size becomes of importance.

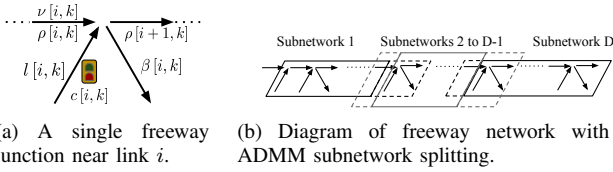
For freeway networks with very small branching factors, it is reasonable to assume the following:

- For each subnetwork, the number of bordering links is *constant*.
- The number of shared state and control variables grows *linearly* with the time-horizon for each subnetwork.
- The number of subnetworks scales linearly with network size (for fixed-size subsystems).

One concludes that the amount of communication required for the A-ADMM subnetwork splitting method would scale linearly with the network size and quadratically with time-horizon length. If we were to instead decompose our system, for instance, across time-slices, the communication requirement would scale quadratically with network size and linearly with time-horizon length. Given our assumption of a fixed time-horizon, the subnetwork splitting approach for network-flow MPC has the added benefit of better scaling in the communication requirements. Additionally, [15] establishes, under strong-convexity assumptions, that weak connectivity in the dependency graph of the multi-agent system leads to faster convergence of ADMM. We conjecture such convergence benefits would extend to the subnetwork splitting approach we outline.

V. DISTRIBUTED, COORDINATED OPTIMAL RAMP METERING AND VSL

We apply distributed optimization via subnetwork splitting to the problem of coordinated, predictive freeway onramp metering and VSL control [22], [4], [16], where traffic lights on freeway onramps are used to regulate the flow entering freeway mainlines and speed limits are dynamically adapted in order to prevent congestion and improve such metrics as driver travel time and speed variability. The term *coordinated* indicates that many traffic lights and VSL signs along a freeway stretch will act cooperatively, given that conditions near one onramp or VSL sign may eventually affect conditions at a neighboring onramp or VSL sign. The term *predictive* indicates that the metering/VSL strategy should anticipate future conditions on the roadway using traffic demand predictions and an underlying model of the evolution of the freeway system.



(a) A single freeway junction near link i . (b) Diagram of freeway network with A-ADMM subnetwork splitting.

Fig. 3. Overview of the freeway ramp metering network and state evolution. Figure 3a shows the dynamical state and control variables of a particular junction i on the freeway. The relation between mainline density $\rho[i, k]$, onramp queues $l[i, k]$, metering control rate $c[i, k]$, VSL $\nu[i, k]$, and boundary condition split ratios $\beta[i, k]$ for a given time-step k are depicted, and mathematically expressed in Equations (15)-(21). Figure 3b shows how one may partition the linear network into subnetworks. While subnetworks may have internal links and onramps, they will also include links and onramps immediately upstream and downstream as part of their shared state (denoted by the dashed-line boxes), giving the appearance of overlapping subnetworks.

Similar to discretized freeway models following the cell transmission model (CTM) approach [17] taken in [18], [23], we adopt the *Link-Node* CTM model presented in [16]. The network is given as a linear sequence of mainline link, onramp and offramp triples², as depicted in Figure 3. We establish the state variables of the system as $s = \{\rho[i, k], l[i, k] : i \in [1, N], k \in [1, T]\}$, where $\rho[i, k]$ is the number of vehicles on the mainline link i (with unit length) and $l[i, k]$ is the number of vehicles queued on onramp i , both at time-step k . Additionally, the control variables are $x = \{(c[i, k], \nu[i, k]) : i \in [1, N], k \in [1, T]\}$, where $c[i, k] \in \mathbb{R}_+$ is the maximum vehicles that can leave onramp i at time k (ramp metering rate), and $\nu[i, k]$ is the maximum speed of vehicles on link i at time k (VSL rate). The following system of equations relate the state of the freeway at time-step $k - 1$ to k :

$$d[i, k] = \min(c[i, k], l[i, k]) \quad (15)$$

$$\sigma[i, k] = \min(w(\rho^{\max} - \rho[i, k]), f^{\max}) \quad (16)$$

$$\delta[i, k] = \min(\nu[i, k]\rho[i, k], f^{\max})(1 - \beta[i, k]) + d[i, k] \quad (17)$$

$$f[i, k] = \min(\nu[i, k]\rho[i, k], f^{\max}) \times \frac{\min(\delta[i, k], \sigma[i + 1, k])}{\delta[i, k]} \quad (18)$$

$$r[i, k] = d[i, k] \frac{\min(\delta[i - 1, k], \sigma[i, k])}{\delta[i - 1, k]} \quad (19)$$

$$l[i, k] = l[i, k - 1] + D[i, k] - r[i, k - 1] \quad (20)$$

$$\rho[i, k] = \rho[i, k - 1] + f[i - 1, k - 1](1 - \beta[i, k - 1]) + r[i, k - 1] - f[i, k - 1], \quad (21)$$

where $w \in \mathbb{R}^+$ is a constant parameter modeling the speed at which congestion propagates backwards. The intermediate variables above all have a physical interpretation: for a time-step k , $d[i, k]$ (resp. $\delta[i, k]$) is the maximum number of vehicles that can exit offramp (resp. mainline cell) i , $\sigma[i, k]$

²Freeway models with more general network topologies exist [20] and allow direct application of the subnetwork splitting method presented herewithin. We limit our discussion to linear freeway networks to simplify the presentation.

is the maximum vehicles that can enter mainline cell i , and $r[i, k]$ (resp. $f[i, k]$) is the actual number of vehicles leaving onramp (resp. mainline cell) i .

The recursive definitions above require an initial condition,

$$s^0 = \{\rho^0[i], l^0[i] : i \in [1, N]\}, \quad (22)$$

and boundary conditions at the left and right extremes of the network,

$$(s^L, s^R) = \{(s^L[k], s^R[k]) : k \in [0, T]\}, \quad (23)$$

both of which are assumed given. Equations (15)-(19) can be seen as intermediate computations required to update the state variables given in Equations (20)-(21), and not explicitly part of the state vector. We note that the offramps are modeled as stateless, infinite-capacity sinks, and thus are only captured through $\beta[i, k]$, the fraction of vehicles which desire to exit offramp i rather than continue to mainline link $i + 1$ at time-step k . A diagram of the state and control variables for a single junction is given in Figure 3a.

The above dynamics are non-convex, but it is shown in [16] that, assuming some maximum velocity V and ramp flow C , if a set of variables satisfy the following linear inequalities and equalities,

$$f[i, k] \leq \min(\rho[i, k]V, f^{\max}) \quad (24)$$

$$f[i, k](1 - \beta[i + 1, k]) + r[i + 1, k] \leq \min(w(\rho^{\max} - \rho[i + 1, k]), f^{\max}) \quad (25)$$

$$r[i, k] \leq \min(C, l[i, k]) \quad (26)$$

Eqns (20) – (21),

then a control c, ν can be constructed such that f, r, ρ, l, c, ν satisfy Equations (15)-(21). Thus, we can employ common convex optimization techniques, such as interior-point methods [19], on the relaxed problem in order to improve sub-objectives during each iteration of the A-ADMM algorithm with a guarantee of convergence to the global optimum. Other non-convex traffic models, such as *second-order* models [20] could be used for more expressivity, but without the convergence guarantee. We omit the explicit c, ν reconstruction procedure and refer the reader to [16] for details.

As an objective, we use *total travel time* (TTT), or the cumulative time spent by all vehicles on the network. An equivalent representation of TTT is to sum the vehicle count across all time-steps and multiply by the time-length of a single time-step. Thus, taking Δt to be the discrete time-step, TTT is mathematically expressed as

$$f_{\text{TTT}} = \Delta t \sum_{i, k} (\rho[i, k] + l[i, k]),$$

and is decomposable across subnetwork splits.

It is clear from the definitions of s and x above that each state variable is a direct function of only the state and control variables of neighboring links at the previous time-step, and as such, can be decomposed using the subnetwork splitting method in Section IV. Figure 3b depicts such a splitting,

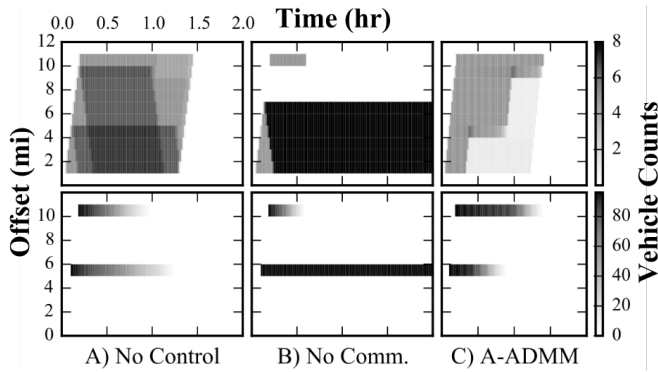


Fig. 4. Space-time diagrams of mainline (row 1) and onramp (row 2) vehicle count evolution for 12 mile network. Large congestion pockets appearing for (A) no control case are reduced using coordinated holding of vehicles on onramps and decreased speed limits during periods of congestion (C). Lack of communication between subsystems (B) leads to an ineffective control policy.

where each subnetwork also includes the neighboring upstream and downstream links as boundary conditions.

The dependency graph (V, E) for such a network has a natural structure, where an edge (i, j) is in E if and only if $j = i + 1$, and thus a subnetwork need only communicate with the linear subnetworks immediately upstream and downstream of itself. Furthermore, only information pertaining to the bordering links and onramps of a subnetwork needs to be shared with its neighbors, allowing a subnetwork to conceal the particular implementation of its internal freeway model from the rest of the system.

VI. NUMERICAL RESULTS

All simulations were run on a personal laptop with a 2.4 GHz Intel Core i5 using 8 GB of RAM. The code was implemented in Scala [24], and makes use of the general-purpose nonlinear optimization software IpOpt [25].

A. Convergence with Number of Subnetworks

We first investigate the numerical convergence of the A-ADMM metering and VSL controller on a model 4-lane freeway network spanning 12 miles ($N = 12$ cells) with 3 onramps and 2 offramps over a 2 hour simulation ($T = 120$ time-steps). We consider three different partitionings by splitting the network into 2, 3 and 4 subnetworks, respectively. We also simulated the following alternative controllers for comparison:

- No control: Metering rates are set to maximum ramp flux rates C and speeds are set to free flow velocity V .
- Centralized: A single optimal control problem over the entire freeway is solved as a convex optimization problem. This solution gives the theoretical lower bound on TTT.
- No communication: Individual subnetworks optimize over their own decomposed TTT objective, with no exchange of information between subnetworks.
- Communicative: Subnetworks iteratively optimize over decomposed objectives and exchange the resulting predicted boundary conditions with neighbors until result-

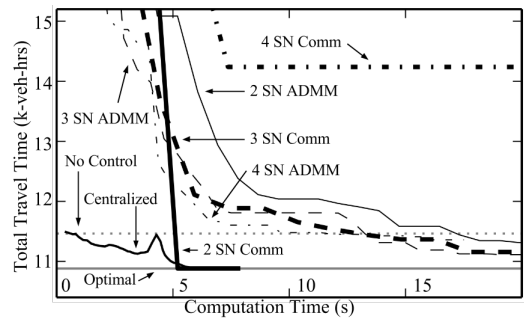
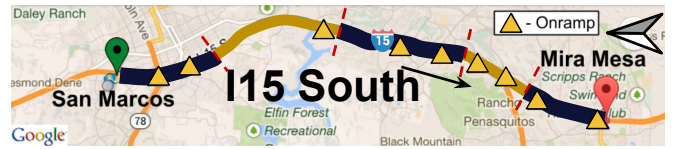


Fig. 5. TTT vs. computation time for several different control schemes and subnetwork (SN) partitionings. The no-communication results are omitted due to poor performance.



(a) Geographical depiction.

Opt.	No Con.	Cent. MPC	No Comm.	Comm.	A-ADMM
2.572	2.605	2.584	4.529	8.453	2.589

(b) TTT summary in 1000 vehicle hours.

Fig. 6. I15 South MPC simulation summary. Figure 6a shows the freeway under consideration partitioned into 5 subnetworks, while Table 6b gives a summary of the performance of the different ramp metering/VSL controllers.

ing boundary conditions converge (see [4]). There is no guarantee of convergence of boundary conditions or of finding the global optimum.

Figure 4 gives a space-time depiction of the mainline and onramp vehicle evolution for the no control, no communication, and A-ADMM controllers.

The convergence results for the 12 mile freeway network are summarized in Figure 5. The centralized approach is faster than the distributed approaches (A-ADMM and communicative) as the former does not require an outer communication loop. As the number of network partitions increases, A-ADMM converges faster to the optimum due to the parallelization of the subnetwork optimizations. Furthermore, the communicative algorithm degrades in performance with increasing number of partitions due to the increase in communication requirements and lack of global objective coordination. If a decentralized algorithm is required for architectural reasons, then the A-ADMM approach is shown to be most desirable due to the lower degree of coordination than the centralized approach and better convergence than the communicative approach.

B. Distributed MPC for I15 Network

MPC simulations were run on a calibrated model of the I15 South freeway in San Diego, CA with boundary flow data taken from measurements recording during a morning rush hour. The simulation spans 20 miles ($N = 32$ cells), contains 9 onramps, 8 offramps and runs over a 170 minute window ($T = 1000$ time-steps) with an MPC update time of 17 minutes and a horizon of 25 minutes. The network is

partitioned into 5 subnetworks and is depicted geographically in Figure 6a.

Travel Time Results: Table 6b gives a summary of the performance of the A-ADMM MPC controller along with other controllers. The optimal TTT is the overall performance lower bound given perfect estimation of initial and boundary conditions and an infinite time horizon and computation time. The centralized, infinite computation time MPC controller is given as a lower-bound on the expected performance under noisy initial and boundary conditions. The results indicate that the A-ADMM controller performs nearly as well as the centralized MPC controller. The communicative and non-communicative approaches were not able to improve upon no control. The communicative approach performed worse than the non-communicative approach because it terminated after reaching a set number iterations on a highly inefficient solution, due to its lack of convergence guarantees.

Running Time: The theoretical optimum was allowed to run until convergence and took approximately 30 minutes to run. For all MPC controller types, each MPC update (every 17 simulation minutes) was allowed to run until either convergence was reached, or until approximately 10 wall-clock minutes was reached to enforce the constraint that the algorithms should be of real-time use.

VII. CONCLUSION

We presented a distributed optimization algorithm based on the multi-agent A-ADMM formulation, applied to systems with both shared control and shared state. We showed how the technique could be used for MPC problems on networked dynamical systems to allow subnetworks to update and communicate in a decentralized and asynchronous manner. We derived a distributed, cooperative freeway ramp metering and VSL controller based on the technique, and ran simulations on a realistic freeway network, which demonstrated improvements in performance over simpler decentralized MPC approaches.

As future work, we will investigate an accelerated [14] version of the A-ADMM algorithm and investigate performance improvements. Additionally, we hope to do sensitivity analysis on models of real traffic networks using the magnitude of the dual parameters (λ) to inform traffic control systems designers on which subnetworks have strong coupling and would benefit from additional coordination.

REFERENCES

- [1] A. M. Bayen, R. Raffard, and C. Tomlin, "Adjoint-based control of a new eulerian network model of air traffic flow," *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 5, pp. 804–818, Sep. 2006.
- [2] A. Kotsialos and M. Papageorgiou, "Efficiency and equity properties of freeway network-wide ramp metering with AMOC," *Transportation Research Part C: Emerging Technologies*, vol. 12, no. 6, pp. 401–420, 2004.
- [3] B. O'Donoghue, G. Stathopoulos, and S. P. Boyd, "A Splitting Method for Optimal Control," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 6, pp. 2432–2442, 2013.
- [4] J. R. D. Frejo and E. F. Camacho, "Feasible Cooperation Based Model Predictive Control for Freeway Traffic Systems," *Conference on Decision and Control*, vol. 50, no. 2, pp. 5965–5970, 2011.
- [5] Y. Pu, M. N. Zeilinger, and C. N. Jones, "Fast Alternating Minimization Algorithm for Model Predictive Control," in *IFAC World Congress*, 2014, p. (To appear).
- [6] P. Giselsson, M. D. Doan, T. Keviczky, B. D. Schutter, and A. Rantzer, "Accelerated gradient methods and dual decomposition in distributed model predictive control," *Automatica*, vol. 49, no. 3, pp. 829–833, 2013.
- [7] C. Chen, Z. Liu, W.-H. Lin, S. Li, and K. Wang, "Distributed modeling in a mapreduce framework for data-driven traffic flow forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 22–33, 2013.
- [8] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel, "Distributed ADMM for model predictive control and congestion control," in *Conference on Decision and Control*, 2012, pp. 5110–5115.
- [9] E. Camponogara and L. B. De Oliveira, "Distributed optimization for model predictive control of linear-dynamic networks," *IEEE Trans. Syst., Man, Cybern. A*, vol. 39, no. 6, pp. 1331–1338, 2009.
- [10] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright, "Distributed MPC strategies with application to power system automatic generation control," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 6, pp. 1192–1206, 2008.
- [11] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.
- [12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.
- [13] E. Wei and A. Ozdaglar, "On the $O(1/k)$ Convergence of Asynchronous Distributed Alternating Direction Method of Multipliers," *arXiv preprint arXiv:1307.8254v1*, p. 30, 2013.
- [14] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," in *Soviet Mathematics Doklady*, vol. 27, no. 2, 1983, pp. 372–376.
- [15] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the admm in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.
- [16] A. Muralidharan and R. Horowitz, "Optimal control of freeway networks based on the link node cell transmission model," in *American Control Conference*. IEEE, 2012, pp. 5769–5774.
- [17] C. F. Daganzo, "The cell transmission model, part II: network traffic," *Transportation Research Part B: Methodological*, vol. 29, no. 2, pp. 79–93, 1995.
- [18] M. L. Delle Monache, J. Reilly, S. Samaranyake, W. Krichene, P. Goatin, and A. M. Bayen, "A PDE-ODE model for a junction with ramp buffer," *SIAM Journal on Applied Mathematics*, vol. 74, no. 1, pp. 22–39, 2014.
- [19] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004, no. 3.
- [20] M. Garavello and B. Piccoli, *Traffic flow on networks*. American institute of mathematical sciences Springfield, MA, USA, 2006, vol. 1.
- [21] M. Gugat, M. Dick, and G. Leugering, "Gas flow in fan-shaped networks: classical solutions and feedback stabilization," *SIAM Journal on Control and Optimization*, vol. 49, no. 5, pp. 2101–2117, 2011.
- [22] M. Papageorgiou, H. Hadj-Salem, and J. Blossville, "ALINEA: A local feedback control law for on-ramp metering," *Transportation Research Record*, vol. 1320, pp. 58–64, 1991.
- [23] J. Reilly, S. Samaranyake, M. L. Delle Monache, W. Krichene, P. Goatin, and A. Bayen, "Adjoint-based optimization on a network of discretized scalar conservation law PDEs with applications to coordinated ramp metering," *Journal of Optimization Theory and Applications (under review)*, 2014.
- [24] "ADMM Ramp Metering-VSL Controller Code," 2014. [Online]. Available: <https://github.com/jackdreilly/LinkNodeOptimizer>
- [25] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.